

ANÁLISE COMPARATIVA DE FERRAMENTAS DE REDES NEURAIS ARTIFICIAIS

Ivan Luis Suptitz (UNISC)

ivansuptitz@gmail.com

Rejane Frozza (UNISC)

frozza@unisc.br

Rolf Fredi Molz (UNISC)

rolf@unisc.br



Este artigo promove uma revisão bibliográfica e um estudo exploratório de ferramentas que possibilitam realizar modelagem de aplicações com a técnica de redes neurais artificiais (RNA), por meio de uma interface gráfica e que possam ser utilizadas para o ensino e compreensão de algoritmos de RNA. São analisados aspectos como licença de uso da ferramenta, presença de interface gráfica, algoritmos disponíveis e finalidade. O objetivo é identificar possíveis lacunas deixadas pelas ferramentas existentes de forma a validar a construção de um sistema que atenda aos requisitos não atendidos, ou atendidos parcialmente.

Palavras-chave: Redes neurais artificiais, ferramentas de modelagem de RNA

1. Introdução

Redes neurais artificiais (RNA) são um artefato da área de inteligência artificial (IA) cada vez mais explorado para resolução de problemas relacionados à classificação e agrupamento de padrões (dados), especialmente no tratamento de imagens, onde a aplicação de algoritmos convencionais não tem bons resultados devido à complexidade das estruturas. Entretanto, o desenvolvimento desta classe de algoritmos não é trivial e, muitas vezes, o próprio ensino de RNA no meio acadêmico é uma tarefa difícil.

Nos últimos anos têm ocorrido a criação e o aperfeiçoamento das linguagens de programação de alto nível, o que alavancou o desenvolvimento de *softwares* e componentes destinados às áreas da inteligência artificial, possibilitando o surgimento de uma diversidade de ferramentas que implementam algoritmos de IA, inclusive RNAs (MEDEIROS, 2006 *apud* DIAS, 2008).

O objetivo desta pesquisa é analisar uma lista de ferramentas que oferecem a possibilidade de especificar modelos de redes neurais de forma que possam ser utilizadas para ensino de RNAs e para criação de aplicações específicas. A modelagem deve ser preferencialmente por meio de uma interface gráfica, mas também serão avaliadas as ferramentas que disponibilizam *Application Programming Interfaces* (APIs), ou bibliotecas de códigos-fonte para apoiar o desenvolvimento de um *software* com as características procuradas.

O artigo está organizado nas seguintes seções: A seção 2 apresenta os principais conceitos de RNAs; a seção 3 descreve o procedimento metodológico executado na pesquisa; a seção 4 apresenta as ferramentas avaliadas; a seção 5 contém as considerações finais.

2. Redes neurais artificiais

Sistemas computacionais são facilmente implementados quando o objetivo é executar processamentos puramente matemáticos e realizam estes procedimentos muito melhor do que seres humanos poderiam realizar. Entretanto, outras tarefas que são consideradas simples para pessoas, como reconhecimento de padrões visuais, por exemplo, são de difícil modelagem em um sistema computacional.

Redes neurais artificiais, comumente chamadas apenas de redes neurais, são sistemas computacionais que têm sua arquitetura inspirada no modo como trabalha o cérebro humano, com o objetivo de tentar reproduzir suas habilidades como o reconhecimento de padrões. Elas são compostas por uma determinada quantidade de entradas, unidades de processamento, ligações entre essas unidades e saídas. As unidades de processamento são denominadas de neurônios, os quais são conectados uns aos outros por meio das ligações (imitando as sinapses que ocorrem em uma rede de neurônios natural). Cada ligação possui um valor de peso, que determina a intensidade da ligação. Quando recebe alguma entrada, o neurônio obtém o valor real do estímulo realizando uma operação chamada função de soma que avalia o valor recebido com os respectivos pesos de cada conexão. Em seguida, o resultado da função de soma é aplicado à função de transferência (ou limiar) que decide se o sinal deve ser encaminhado à etapa seguinte ou não, conforme os parâmetros do sistema (HAYKIN, 1999).

As RNAs podem ser classificadas quanto a sua topologia em: redes diretas (também chamada de *FeedForward*) nas quais um sinal é aplicado à entrada e percorre em uma única direção através das camadas intermediárias, ou ocultas, até a saída; e redes com ciclos, nas quais ocorre realimentação do sinal de saída para entrada (RAO, 1993).

Quanto à forma de aprendizado, as RNA podem ser classificadas em: aprendizado supervisionado, no qual a rede é condicionada a reconhecer determinados padrões mediante a apresentação prévia de um grupo de treinamento; e aprendizado não supervisionado, onde a rede opera sobre um conjunto de dados de entrada sem um treinamento prévio e fornece na saída o agrupamento dos padrões mais semelhantes. Existem diversos algoritmos de aprendizado na literatura para ambas as formas de aprendizado, o mais conhecido é o *Backpropagation* que é em um algoritmo supervisionado aplicado em redes *FeedForward*. De acordo com Rao (1993) 80% de todos os projetos de redes neurais desenvolvidos utilizam o

algoritmo de *Backpropagation*. O aprendizado é construído após a apresentação de cada padrão por meio da avaliação do erro produzido na saída. É calculado um fator de correção que é utilizado para ajustar os pesos das conexões entre os neurônios (RAO, 1993).

Resta citar que existem diversos modelos de RNA implementados. Representando a classe de aprendizado supervisionado, o modelo mais comum é o *perceptron* que geralmente utiliza o algoritmo *Backpropagation* para treinamento supervisionado. Na classe das RNAs de aprendizado não supervisionado o modelo de Kohonen é o mais comum, sendo também conhecido pela expressão de *Self-Organizing Map* (SOM) que é capaz de reconhecer relações entre os padrões apresentados e realizar agrupamentos (KOHONEN, 2001).

Neste trabalho o modelo *perceptron* é chamado de *Multi Layer Perceptron* (MLP), que designa uma construção deste modelo em mais de duas camadas. O Modelo de Kohonen é referido por SOM. Outros algoritmos ou modelos menos comuns têm sua descrição por extenso.

3. Metodologia da pesquisa

O presente artigo se propõe a fazer um levantamento de algumas ferramentas para modelagem de RNA. Por esse motivo a metodologia de pesquisa é constituída essencialmente por uma busca exploratória por Ferramentas de RNA que serão comparadas, a partir de um conjunto de critérios, para análise. O artigo também descreve esta análise.

Inicialmente foi realizada uma busca no portal de periódicos “Capes” (www.periodicos.capes.gov.br) por artigos que citam redes neurais, modelagem de redes neurais ou ferramentas de redes neurais. O material encontrado serviu como base para a revisão bibliográfica e para encontrar referências de ferramentas que possivelmente atenderiam aos requisitos desejados. Foram utilizados livros e periódicos nacionais e internacionais, acessados através do portal de periódicos “Capes” nas bases de dados *Scopus* e *Web of Science*. Na busca em ambas as bases, para os campos título, resumo e palavras chave, foram utilizadas as seguintes palavras chave: “*modeling*”, “*artificial neural network*” e “*tool*”. Foram filtrados apenas artigos, nos idiomas inglês e português, nas áreas de “*Engineering*” e “*Computer Science*”.

Também foi realizada uma procura por *softwares* de RNA diretamente na internet por meio da ferramenta de busca Google (www.google.com), em inglês e português.

Dentre a lista de ferramentas encontradas, foram eliminadas as que não permitiram acesso livre para *download* e instalação, seja por causa de exigência de compra de licença, ou por indisponibilidade do *link* de *download*.

Os critérios estabelecidos para a avaliação das ferramentas foram os seguintes:

- Licença de uso (se tem uso gratuito ou não);
- Modelos e algoritmos de RNA disponíveis;
- Disponibilidade de modelagem por interface gráfica;
- Linguagens de programação na qual a ferramenta disponibiliza API para uso por parte de desenvolvedores de *software*; e
- Propósito da ferramenta.

4. Ferramentas avaliadas

A pesquisa realizada, com base nos critérios citados, resultou na seguinte lista de ferramentas analisadas:

- Accord.NET (SOUZA, 2015)
- AForge.NET (KIRILLOV, 2015)
- ADReNA: Ambiente de desenvolvimento de aplicações em redes neurais artificiais (KIST, 2013)
- EasyNN-plus (EASYN, 2015)
- Encog: Encog Machine Learning Framework (HEATON, 2010)
- FANN: Fast Artificial Neural Network Library (NISSEN, 2015)
- IDRISI (CLARK LABS, 2015)
- NEST: Neural Simulation Technology Initiative (GEWALTIG & DIESMANN, 2007)
- NeuronDotNet: Neural Networks in C# (NEURONDOTNET, 2015)
- Neuroph: Java Neural Network Framework (NEUROPH, 2015)
- OpenNN: An Open Source Neural Networks C++ Library (LOPEZ, 2014)
- WEKA: Waikato Environment for Knowledge Analysis (HALL *et al.*, 2009)

A ferramenta Accord.NET é um *framework* de código aberto escrito na linguagem C#, construído sobre o AForge.NET (que também foi analisado) e implementa diversos algoritmos de aprendizado de máquina, inclusive de redes neurais, juntamente com bibliotecas de áudio e processamento de imagem. Ela tem por finalidade fornecer a estrutura necessária para a construção de visão e audição computacional, processamento de sinal e aplicações estatísticas. Apesar de ser uma biblioteca de código, disponibiliza 47 exemplos prontos para uso, juntamente com os respectivos códigos fonte. Possui uma documentação bastante detalhada disponibilizada sob a forma de uma *wiki*.

ADReNA é um *software* para apoiar o desenvolvimento de dois modelos de redes neurais artificiais. A ferramenta disponibiliza uma interface para criar modelos *perceptron* e de Kohonen, bem como para salvar, carregar e executar estes modelos. Também disponibiliza uma API para desenvolvimento de aplicações nas linguagens C# e Java.

AForge.NET é um *framework* escrito na linguagem C# e também tem o código aberto. É projetado para desenvolvedores e pesquisadores utilizarem nas áreas de Visão Computacional

e IA. Possibilita trabalhar com processamento de imagens, redes neurais, algoritmos genéticos, lógica *fuzzy*, aprendizagem de máquina e robótica.

EasyNN-plus é um *software* comercial desenvolvido pela empresa “*Neural Planner Software*”. O objetivo é modelar uma RNA multicamadas, treinar a rede com matrizes de dados que podem ser importados de arquivos texto, planilhas e imagens, e, em seguida, efetuar consultas de predição sobre o modelo criado. De acordo com a página web da empresa, o *software* é utilizado para fazer previsões em negociação de ações, investimentos, seguros, mercado imobiliário, esportes, previsão do tempo, entre outros. O aplicativo é disponibilizado com uma licença de avaliação para um período de 30 dias, após o qual é necessário comprar uma licença permanente. A mesma companhia também disponibiliza uma versão da ferramenta de uso gratuito sob o nome de “JustNN”, porém com limitações de escala de uso.

Encog é um *framework* em desenvolvimento desde 2008. Estão implementados algoritmos de aprendizado de máquina, Redes Neurais Artificiais, Algoritmos Genéticos, Redes Bayesianas, entre outros. De acordo com o autor, os algoritmos implementados são altamente escaláveis, operando com *Multi-Threads*. Além disso, pode fazer uso do processador da placa gráfica para melhorar os tempos de processamento. Como as bibliotecas de código requerem um elevado conhecimento de programação, é disponibilizada uma interface gráfica por meio do *software* “*Encog Workbench*” com o objetivo de facilitar o uso das funções da API.

FANN é uma biblioteca *open source* que implementa uma estrutura genérica de redes neurais em linguagem de programação C. Tem como objetivo disponibilizar um núcleo robusto para instânciação de RNA em 25 diferentes linguagens de programação (C#, Java, C++, node.js, PHP, Python, D, MQL4, Perl, Ruby, Delphi, Tcl, Lua, Visual Prolog 7, SWI Prolog, Go, Soap, Matlab, R, Ada, Haskell, Grass, Octave, Squeark Smalltalk e Pure Data). A biblioteca não oferece uma implementação pronta de um modelo de RNA, ao invés disso o desenvolvedor deve especificar o número de camadas, a conexão entre os neurônios e o algoritmo de aprendizado.

IDRISI é um *software* de licença paga utilizado para geoprocessamento, portanto, ele é muito específico para processamento de fotos de relevo. Pode ser utilizada gratuitamente por um período de avaliação de 30 dias. A ferramenta oferece diversos algoritmos para fazer processamento de imagens, entre eles o MLP e o SOM ambos atuando como classificadores

de imagens. Ela trabalha com um formato de dados próprio das imagens (basicamente são convertidas para uma matriz de números inteiros). Todos os algoritmos são manipulados por meio de interface gráfica. Não é disponibilizada API para desenvolvimento.

NEST é um *software* de código aberto construído para simular redes neurais semelhantes às do cérebro humano. Diferentemente de outras ferramentas de RNA que possuem morfologia bem definida de neurônios e conexões, o NEST suporta mescla de diferentes tipos de neurônio e de sinapses na rede, pois o foco está na estrutura da rede como um todo. O programa roda em uma *Shell* com comandos para o interpretador Python. O usuário precisa descrever a rede nesta interface e iniciar a execução da simulação.

NeuronDotNet é uma API de código aberto escrita na linguagem C#. O objetivo do projeto é facilitar o desenvolvimento de aplicações de RNA fornecendo uma estrutura centralizada para a criação, treinamento e uso de diferentes tipos de redes neurais artificiais.

Neuroph é um ambiente de desenvolvimento integrado (IDE) específico para desenvolver os modelos de RNA mais comuns. É disponibilizado juntamente com uma biblioteca de código aberto, escrita em Java. Possui ainda uma ferramenta gráfica que permite criar e salvar os modelos de RNA. Apresenta boa documentação *on-line*.

OpenNN (em uma versão antiga tinha o nome de Flood) é uma biblioteca C++ específica para o modelo MLP. Apresenta diversos algoritmos de treinamento e uma boa documentação. O *software* é desenvolvido desde 2008. A última versão é de maio de 2014.

O Weka é uma plataforma livre para mineração de dados bastante conhecida. Foi desenvolvida pela Universidade Waikato na Nova Zelândia. O projeto iniciou em 1992, tendo a primeira *release* pública disponibilizada em 1996. O *software* era inicialmente desenvolvido na linguagem C, sendo que posteriormente foi migrado para Java. Atualmente encontra-se na versão 3.6.12. A ferramenta contém algoritmos de inteligência artificial para processamento de dados, classificação, regressão, regras de associação, segmentação e agrupamento. Entre a coleção de algoritmos disponível está uma implementação do MLP e de SOM. Os algoritmos podem ser aplicados diretamente no conjunto de dados através da própria interface da ferramenta ou podem ser integrados a outro sistema através de chamadas por código Java. Por se tratar de uma ferramenta voltada para mineração de dados, a interface gráfica não disponibiliza opção de modelagem de RNA, somente escolha do algoritmo.

Na tabela 1, as ferramentas pesquisadas estão relacionadas juntamente com os critérios avaliados.

Tabela 1 - Relação de ferramentas pesquisadas

Nome da ferramenta	Gratuita	Modelos e algoritmos disponíveis	Permite modelagem gráfica	Linguagens em que fornece API	Propósito
Accord.NET	Sim	Deep Belief Network; Restricted Boltzmann Machine	Não	C#	API para desenvolvimento de visão computacional
ADReNA	Sim	MLP e SOM	Sim	C#; Java	Modelagem de RNA para fim didático
AForge.NET	Sim	MLP; SOM; Elastic network	Não	C#	API para desenvolvimento de visão computacional
EasyNN-plus	Não	MLP	Sim	-	Modelagem de RNAs para predições
Encog	Sim	ADALINE; MLP; Hopfield; SOM	Sim	Java; .NET; C++	API para desenvolvimento de sistemas de IA
FANN	Sim	MLP	Não	Java; C#; PHP; C++; Python; Ruby; Delphi; Lua (e mais 17)	API para desenvolvimento de RNA rápidas em várias linguagens
IDRISI	Não	MLP; SOM	Sim	-	Geoprocessamento
NEST	Sim	Não implementa modelo específico	Não	Python	Simular o cérebro humano
NeuronDotNet	Sim	MLP; SOM	Não	C#	API para desenvolvimento de RNA
Neuroph	Sim	ADALINE; MLP; Hopfield; SOM	Sim	Java	API para desenvolvimento de RNA
OpenNN	Sim	MLP	Não	C++	Fornecer biblioteca de código para desenvolvimento
WEKA	Sim	MLP; SOM	Não	Java	Mineração de dados

Fonte: dos autores (2015)

Os critérios analisados permitem conhecer o que as ferramentas oferecem e o que ainda necessitam para se tornarem cada vez mais úteis para o desenvolvimento de diferentes aplicações.

5. Discussões e considerações finais

Da lista de ferramentas levantadas, somente ADReNA, EasyNN-plus, Encog, IDRISI e Neuroph atendem o requisito de disponibilizar uma interface gráfica para modelagem de uma RNA. Atendendo o requisito de gratuidade do *software* é preciso eliminar a ferramenta IDRISI pelo fato de ser paga, e substituir o *software* EasyNN-plus pela versão gratuita JustNN.

Desta forma, restam os programas ADReNA, JustNN, Encog e Neuroph. Todas elas atendem aos requisitos estabelecidos e podem ser utilizadas para ensino de RNA. Entretanto, foram identificadas duas lacunas em comum às quatro ferramentas:

- a) Não existe uma forma fácil de efetuar a entrada dos dados e validar o modelo construído: os dados precisam ser digitados manualmente ou então importados e já pré-processados (prontos para análise da RNA). Para classificar um conjunto de imagens, por exemplo, é preciso recorrer a outras ferramentas para realizar a conversão para a matriz de dados.
- b) Não existe uma forma de visualizar passo-a-passo a execução do modelo, com possibilidade de observar o ajuste dos pesos, acompanhar a execução das funções de soma e transferência dentro dos neurônios.

Ambas as características são consideradas importantes para melhorar a curva de aprendizado acadêmico sobre RNAs. Justifica-se desta forma a construção de uma ferramenta que além de possibilitar modelagem de redes neurais em uma interface gráfica e de ser gratuita, disponha de uma interface de aquisição de dados, possibilitando entrada de dados brutos, fornecendo opções de pré-processamento e também uma opção de execução do modelo passo-a-passo com visualização das alterações dos dados dentro dos neurônios e nas conexões.

Identificado essa lacuna em aplicativos de modelagem RNA tem-se que, como salientado anteriormente, um conjunto de APIs de código aberto que podem auxiliar a implementação de uma ferramenta de modelagem com os requisitos esperados. Neste sentido, 10 das 12 ferramentas avaliadas disponibilizam APIs, ou bibliotecas de código para auxiliar o desenvolvimento de aplicações com o uso da ferramenta. Na tabela 2, pode-se visualizar a relação das 10 ferramentas com as linguagens nas quais elas disponibilizam APIs.

Tabela 2 - Linguagens das APIs das ferramentas analisadas

Ferramenta	C#	Java	C++	Outras
Accord.NET	Sim	Não	Não	Não
ADReNA	Sim	Sim	Não	Não
AForge.NET	Sim	Não	Não	Não
Encog	Sim	Sim	Sim	Não
FANN	Sim	Sim	Sim	Sim
NEST	Não	Não	Não	Sim
NeuronDotNet	Sim	Não	Não	Não
Neuroph	Não	Sim	Não	Sim
OpenNN	Não	Não	Sim	Não
WEKA	Não	Sim	Não	Não

Fonte: dos autores (2015)

Como se pode observar, 6 das 10 ferramentas disponibilizam bibliotecas na linguagem C#, 4 em Java, 3 em C++ e outras 2 em outras linguagens diversas. Todas elas podem contribuir de alguma forma para a construção de uma ferramenta com as características buscadas. Tendo em vista o número maior de bibliotecas disponíveis e melhor qualidade de documentação, recomenda-se a construção da ferramenta proposta na linguagem C# com as APIs AForge.NET e, ou Accord.NET.

REFERÊNCIAS

CLARK LABS, **IDRISI**, Clark University 950 Main Street, Worcester MA USA. Disponível em <<http://www.clarklabs.org>>. Acessado em 25 de abril de 2015.

DIAS, C. G. **A implementação e o estudo de redes neurais artificiais em ferramentas de software comerciais**. Exacta, São Paulo, v. 6, n. 1, p. 119-128, jan./jun. 2008.

EASNN. Disponível em <<http://www.easynn.com>>. Acessado em 18 de abril de 2015.

GEWALTIG M-O &DIESMANN M (2007) NEST (**Neural Simulation Tool**) Scholarpedia 2(4):1430.

HALL, M. *et all*. **The WEKA Data Mining Software: An Update**; SIGKDD Explorations, Volume 11, Issue 1.2009.

HAYKIN, S. **Neural Networks. A Comprehensive Foundation**. 2ª ed. Prentice Hall, New Jersey, USA, 1999.

HEATON, J. **Encog java and dotnet neural network framework**. Heaton Research, 2010. Disponível em <<http://www.heatonresearch.com/encog>>. Acessado em 18 de abril de 2015.

KIRILLOV, A. **AForge.NET Framework**. Disponível em <<http://www.aforgenet.com>>. Acessado em 18 de abril de 2015.

KOHONEN, Teuvo. **Self-Organizing Maps** – 3ª ed. Springer-Verlag, Berlin Heidelberg, 2001.

KIST, M. H. **ADReNA: Ambiente de desenvolvimento de aplicações em redes neurais artificiais**. Trabalho de Conclusão de Curso, UNISC, 2013.

LOPEZ, R. **Open NN: An Open Source Neural Networks C++ Library**, 2014. Disponível em <<http://www.cimne.com/flood>>. Acessado em 25 de abril de 2015.

MEDEIROS, L. F. **Redes neurais em Delphi**. 2 ed. Florianópolis: Visual Books, 2006.

NEURONDOTNET. **Neural Networks in C#**. Disponível em <<http://sourceforge.net/projects/neurondotnet>>. Acessado em 25 de abril de 2015.

NEUROPH: **Java Neural Network Framework**. Disponível em <<http://neuroph.sourceforge.net>>. Acessado em 18 de abril de 2015.

NISSEN, S. **Fast Artificial Neural Network Library**. Disponível em <<http://leenissen.dk/fann>>. Acessado em 25 de abril de 2015.

RAO, Valluru B.; RAO, Hayagriva V. **C++ Neural Networks and Fuzzy Logic**. MIS:Press, New York, USA, 1993.

SOUZA, C. R. **The Accord.NET Framework**. Disponível em <<http://accord-framework.net>>. Acessado em 18 de abril de 2015.